# DAITC
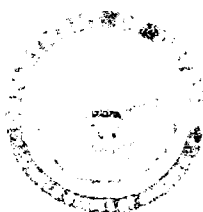
# The DoD Gateway Information System (DGIS): The Development Toward Artificial Intelligence and Hypermedia in Common Command Language

## DAITC/TR-88/009

Allan D. Kuhn

December 1988

DTIC
ELECTE
FEB 16 1989
S    α H    D

AD-A203 674

89    2  16  003

## REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

| 1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS |
|---|---|

| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION/AVAILABILITY OF REPORT |
|---|---|
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | Approved for public release; distribution unlimited |

| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) | 5. MONITORING ORGANIZATION REPORT NUMBER(S) |
|---|---|
| DAITC/TR-88/009 | DTIC/TR-89/4 |

| 6a. NAME OF PERFORMING ORGANIZATION Defense Applied Information Technology Center | 6b. OFFICE SYMBOL (If applicable) | 7a. NAME OF MONITORING ORGANIZATION Defense Technical Information Center |
|---|---|---|
| 6c. ADDRESS (City, State, and ZIP Code) 1800 N. Beauregard Street Alexandria, VA 22311-1784 | | 7b. ADDRESS (City, State, and ZIP Code) Cameron Station Alexandria, VA 22304-6145 |

| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION | 8b. OFFICE SYMBOL (If applicable) | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |
|---|---|---|

| 8c. ADDRESS (City, State, and ZIP Code) | 10. SOURCE OF FUNDING NUMBERS | | | |
|---|---|---|---|---|
| | PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO. | WORK UNIT ACCESSION NO. |
| | | | | |

11. TITLE (Include Security Classification)
DoD Gateway Information System (DGIS): The Development Toward Artificial Intelligence and Hypermedia in Common Command Language

12. PERSONAL AUTHOR(S)
Allan D. Kuhn

| 13a. TYPE OF REPORT | 13b. TIME COVERED FROM _____ TO _____ | 14. DATE OF REPORT (Year, Month, Day) 8812 | 15. PAGE COUNT 19 |
|---|---|---|---|

16. SUPPLEMENTARY NOTATION
Presented at ONLINE INFORMATION 88, 12th International Online Information Meeting, London, England, 6-8 Dec 88

| 17 COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB-GROUP | Common Command Language, CCL, C language, Artificial Intelligence, PROLOG, Hypermedia, DoD Gateway Information System, DGIS |
| 5 | 2 | | |
| 12 | 5 | | |

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
DGIS CCL Development began with C language prototyping to study its feasibility and issues. The study raised the potential of employing Artificial Intelligence, and DGIS CCL is now proceeding on two levels. The first level is incorporating Artificial Intelligence as a means to handle the heterogeneous universe of diverse information systems. We are doing this with PROLOG, using its logic programming capabilities for structuring knowledge base and blackboard architectures. The knowledge bases contain information about the target system commands and operating characteristics, and about users to establish a more graceful and human response in the human-machine interface. AI-based CCL, still in development, is available to all DGIS users. The second level incorporates the hypermedia capabilities of bit-map systems, to run concurrently with those that are ASCII-dependent. A desktop environment is being developed that includes icon selection, windowing technology, simultaneous query invoking and results display n

(continued on reverse)

| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS | 21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED |
|---|---|
| 22a. NAME OF RESPONSIBLE INDIVIDUAL Allan D. Kuhn | 22b. TELEPHONE (Include Area Code) (202) 274-5367 | 22c. OFFICE SYMBOL DTIC-EB |

DD Form 1473, JUN 86    Previous editions are obsolete.    SECURITY CLASSIFICATION OF THIS PAGE

19.  (Continued)

multiple target systems, and color as an enhancing feature.  As a part of
the hypermedia application, integration with media peripherals such as
CD-ROM databases will also be included.

# DOD GATEWAY INFORMATION SYSTEM (DGIS):
# THE DEVELOPMENT TOWARD ARTIFICIAL INTELLIGENCE AND HYPERMEDIA
# IN COMMON COMMAND LANGUAGE

Allan D. Kuhn

# DOD GATEWAY INFORMATION SYSTEM (DGIS): THE DEVELOPMENT TOWARD ARTIFICIAL INTELLIGENCE AND HYPERMEDIA IN COMMON COMMAND LANGUAGE

*A.D. Kuhn, Defense Technical Information Center (DTIC), USA*

Abstract:  DGIS CCL development began with C language prototyping to
study its feasibility and issues.  The study raised the potential of
employing Artificial Intelligence, and DGIS CCL is now proceeding on
two levels.  The first level is incorporating Artificial Intelligence
as a means to handle the heterogeneous universe of diverse information
systems.  We are doing this with PROLOG, using its  logic programming
capabilities for structuring knowledge base and blackboard
architectures.  The knowledge bases contain information about the
target system commands and operating characteristics, and about users
to establish a more graceful and human response in the human-machine
interface.  AI-based CCL, still in development, is available to all
DGIS users.  The second level incorporates the hypermedia capabilities
of bit-map systems, to run concurrently with those that are ASCII-
dependent.  A desktop environment is being developed that includes
icon selection, windowing technology, simultaneous query invoking and
results display in multiple target systems, and color as an enhancing
feature.  As a part of the hypermedia application, integration with
media peripherals such as CD-ROM databases will also be included.

## I. INTRODUCTION

The Defense Technical Information Center (DTIC) has sponsored
development of the DoD Gateway Information System (DGIS) since 1982, and
installed its first DGIS computer in 1986.  The purpose of DGIS is to
provide online, streamlined methods for identifying, accessing and
analysing information from heterogeneous databases of interest to the DoD
community [CGA85].  The following figure is the top menu of DGIS, and shows
its core operations to achieve this purpose [KAD86]:

```
0        WELCOME TO THE DoD GATEWAY INFORMATION SYSTEM


>>>>>>>>>>INFORMATION TRANSFER MODULES
1        directory       DGIS Directory of Online Resources.
2        communicate     Connect to Information resources and people
3        process         Information product tailoring.


>>>>>>>>>>INFORMATION UTILITIES
4        em              Electronic Mail.
5        files           File operations.


>>>>>>>>>>SUPPORT INFORMATION
6        help            Description of features.
7        users           DGIS registered users
8        info            DGIS news and information.


DGIS HOTLINE NUMBER  (703) 276-8182
or send questions via DGIS EM to 'dgishelp.'

Enter a  menu  number,  a command, "b" to backup, "T" for top, or "e" to end.
   .
```

The core components of DGIS that support its purpose are 'directory', 'communicate', and 'process'. The component of concern for accessing and retrieving is 'communicate', where DGIS users can be automatically connected with and logged into an information system. Once in a system, however, the user has to be knowledgeable of the system to retrieve from it. One of the barriers to searching diverse databases is the lack of a common command language set for retrieving in heterogeneous databases. With the capability to easily access a multiplicity of information sources, the need to eliminate learning multiple native command languages is no longer an issue, but an absolute requirement. To overcome the barrier, DTIC has been developing a Common Command Language (CCL) capability since 1986, first with syntactical C language prototypes to study the feasibility and problems of CCL, and then following on with Artificial Intelligence applications.

## II. OUR FIRST PROTOTYPING DEVELOPMENTS

Our initial effort to implement CCL was motivated by beginning with a simple approach, and trying to get several prototypes up quickly. We adopted the draft standard for common commands prepared by the National Information Standards Organization (NISO) [NISO86/87], of the United States. Programming was done in C, merged with two UNIX utilities that were immediately adaptable to CCL needs. Those two utilities were LEX (generator of lexical analysis programs), and YACC (Yet Another Compiler-Compiler) [UNIXol]. LEX was used for lexical analysis of the CCL prototype C programs, YACC for the syntactical analysis. C was used to implement all remaining semantic processing and miscellaneous tasks [TDTpip].

Communications was highly critical. DGIS had NAM (Network Access Machine) software agents available in the DGIS software for connecting users to databases for native language searching. NAM provided programming for:

a.  Establishing the connection.
b.  Validating user access.
c.  Logging on to the target database, including entry of the logon codes. The NAM agent was reviewed and found adaptable to CCL for communicating the command and response in searching the remote database

system [TDTpip].

Our first prototype was CCL for DIALOG. DIALOG was chosen because it was a system with which many users in the DoD community are familiar, and find easy to use.

The DIALOG prototype was followed by BRS, NASA-RECON, and DROLS in fairly rapid succession. BRS was chosen because it was another major vendor system with many databases, NASA-RECON because it was a Federal government database system, and finally DROLS, DTIC's database system.

III. FIRST PROTOTYPE RESULTS

We terminated C-programming with completion of the four prototypes. The experience we gained was immeasurably useful. The following issues and features resulted from this first prototyping:

1. The Adaptation of the NAM Connection Agent: As mentioned NAM software for connecting with remote databases was already available. Once the sign-on is completed, the user is connected directly with the database. The user would then invoke the CCL translator.

2. CCL Invocation: Once one had accessed a database system through the NAM connection agent, one may invoke the CCL translator with a special key.

3. CCL Translators: The creation of prototype CCL translators taught us that each information system is individualistic and must be treated as such. The translator programming was totally dependent on command mapping requirements for each system. The programmer must also detect anything "hidden" in the target database system that is needed for a response. The CCL translator was a filter that, once activated, intercepted all CCL commands from the user, translated the command, sent the translation (i.e., the target database native command) for execution, and brought the results back to the user [TDTpip]. The translator was deactivated by the conventional <CNTL>d (exit from a process).

4. Native Command Language Option: The option to use the native command language was necessary when we were prototyping only a selected set of commonly used commands. The entry of a native command was made very simple: at the CCL prompt, one precedes the native command with a backward slash (\) to tell the
translator that the native command is coming, e.g.:

        CCL >  \s      (for DIALOG 'select')

5. The CCL Prompt: The prompt 'CCL >' was incorporated as a reminder to the user that one has invoked the CCL utility.

6. CCL Command Verification: When the user invoked a common command, the translation of the invocation was echoed in the database system structure, e.g., for DROLS:

        CCL > find artificial intelligence and psychology

    (echo)      @str@
                artificial intelligence
                and
                psychology
                end

        CCL: Searching...

693

The echo could also be turned off with the command: CCL> noecho .

7. Online Documentation: The HELP feature to show the user how to use the CCL. The documentation, in very abbreviated form, covered the CCL commands available. For example (DIALOG):

```
           CCL > help find

                    CCL format
                          find <term> ...
                    DIALOG2 format
                          select <term> ...
                    DESCRIPTION
                          Initiate a search.
```

8. Shell Spawning while in CCL: We incorporated the capability to exploit a UNIX shell, file, or utility while in the CCL. Use of the capability was at the user's discretion; for example, the user might want to list one's files as a review measure while searching a database. The signal to the CCL translator was an initial bang (!), e.g.,

```
           CCL > !ls      (for listing files)
           CCL > !w       (for seeing who is on the system)
                etc.
```

9. New Commands: In developing the DGIS CCL we found that the NISO standard did not cover several items that we deemed useful. Usefulness was based on the following:

    a.  Functions, prevalent in systems, that aided the user; an example is successive session cost display.

    b.  Functions, not prevalent, seen as highly useful; e.g., listing the accession numbers of finds.

    c.  Functions that we found were needed for an operative CCL; an example is cancelling the translation echo display at one's discretion.

The non-NISO commands that we incorporated under the first prototyping were:

| | |
|---|---|
| COMBINE | Do Boolean operations (and, or, not) on previously created sets. |
| COST | Display session cost thus far. |
| EXECUTE | Execute a previously saved search strategy (in target database). |
| LIST | List accession numbers of search results. |
| NOECHO | Cancel native command function echo to CCL command function. |

10. NISO Standard Common Commands Incorporated in the First Prototyping: As we developed the four prototypes, we included the following commands to enhance the prototype capabilities: CHOOSE, HELP, FIND, DISPLAY, RELATE, FORWARD, and BACK. START and STOP are taken care of by the DGIS automatic connect and disconnect.

11. CCL System Menu Development: As we progressed through the four prototypes, the more unfamiliar the database systems became. The programmer in particular was totally unfamiliar with DROLS. This is a normal situation because DROLS, in addition to being a terse, no-assist system, is a closed system with a relatively small, registered community.

694

It was necessary to ascertain particular DROLS system processes and accommodate them in the CCL programming.

The very terseness of DROLS (including that lack of a prompt) generated the need to experiment with menu sets to step the unfamiliar user through the database. These menus, basically, provide functional information the expert DROLS searcher knows, but is not on the system. CCL menu examples are:

When invoking CCL CHOOSE in DROLS without designating which database -

```
CCL > choose

 Select one of the following files :
         1. Current Reports
         2. Technical Reports
         3. New Accessions
         4. Work Units
 Please enter your choice (1-4) --> 2

 Technical Reports file is selected.

 CCL > find ...(etc.)
```

When invoking CCL DISPLAY for search results in DROLS without designating a display format -

```
CCL > display

Select data type to be displayed:
        1.   Search Results.
        2.   Qualified Results.
        3.   User File.
        4.   Single Technical Report Number.
        5.   Single Current File Number.
        6.   Single Work Unit Number.
        7.   Available Files.
        8.   Information Log.
        9.   Order Log.
        10. Inverted File.

Please enter your choice (1-10) --> 1
Please enter a field no (0 for end of field list) --> 3
Please enter a field no (0 for end of field list) --> 21
Please enter a field no (0 for end of field list) --> 23
Please enter a field no (0 for end of field list) --> 0
```

```
CCL > (Sub command for display mode)

Select a display mode :
         1.      Item by item display.
         2.      Continuous display.
Please enter your choice (1-2) --> 1

--   1  OF   20
-- 1 -  AD NUMBER: P003929
-- 3 -     (etc.)
```

The inclusion of the menu sets aids the CCL user to navigate the unfamiliar system, and hopefully helps eliminate the need to totally rely on user manuals.

695

## IV. MAJOR PROBLEM

Each prototype raised issues and problems which we used to refine the successive prototype. As the prototypes progressed, various problems in working with them led to solutions such as HELP features and menus as mentioned above.

The major problem, however, surfaced as a result of our cumulative experience. We learned that creating "Common Command Language" was NOT a panacea. Programming a "standard" command language was in actuality only substituting one command language for another.

This was most apparent when the DISPLAY function is employed. Quite factually, if the user does not know the DISPLAY formats of an unfamiliar system, one cannot see results. A command with less serious consequences is the FIND function. Using FIND, the user is very likely to be able to enter the query and foment results. But any function involving a display is likely to be dead-ended in no display. This situation simply does not obviate the need for referral to a system's user documentation, which gives instruction in terms of its native command language.

Another example is the CHOOSE function. Some systems identify databases by number, others by acronym. For BRS, one must enter CHOOSE NTIS; in DIALOG, CHOOSE 6. The hydra of options and formats keeps cropping up. Each system must be addressed individually, with the goal of having some central pattern program to draw upon. The crutch we used for the C language-based CCL prototype is the menu.

The creation of a CCL is only one component of the "CCL-need" issue. A second component is creation of a CCL System that allows a user to search in unfamiliar database systems without needing to know that system's operating characteristics. A third component is identifying the critical purposes that a CCL system is to serve.

In the case of DGIS, the criteria for CCL purposes are the DGIS information processing operations, particularly in postprocessing downloaded files. A DGIS postprocessing requirement is to have a tagged citation for translation. Downloaded citations must be translated into the DGIS standard citation format before the automated processing routines can be applied.

This requirement is an example of a criterion for a DGIS CCL system. The CCL system must include function default results for those users unfamiliar with a database, particularly for DISPLAY. The default, on simple invocation of DISPLAY, will provide the fully tagged citation. Additional elements, such as menus and question prompts, e.g., "DISPLAY on last set? y/n," must also be incorporated.

The case of CHOOSE represents another problem environment. In DGIS the solution is the eventual integration of CCL with a Directory of Online Resources. When this is accomplished, the query will be forwarded automatically to the relevant databases through CCL.

The real demon for CCL has turned out to be the idiosyncratic operating characteristics of each information system. The need was the creation of a DGIS Common Command Language System (CCLS), encompassing all the idiosyncracies of all information systems.


## V. THE DECISION FOR ARTIFICIAL INTELLIGENCE

# 1. THE CAUSE

The rigidity and constraints of a straight algorithmic program-based CCL discovered during the prototypings led us to exploring the potential of artificial intelligence. The natural language and expert system possibilities of AI were very appealing. The project's programming technical expert reviewed the main AI programming languages. His recommendation was to explore AI applications with PROLOG, a simple but powerful relational programming language based on the idea of programming in logic [BA-86].

The initial technical reasons for selecting PROLOG were [TDTpip]:

a. The Reversibility of PROLOG: In determining object relationships, a program can be written establishing those relationships, with the inverse of the relationship inherent in the program.
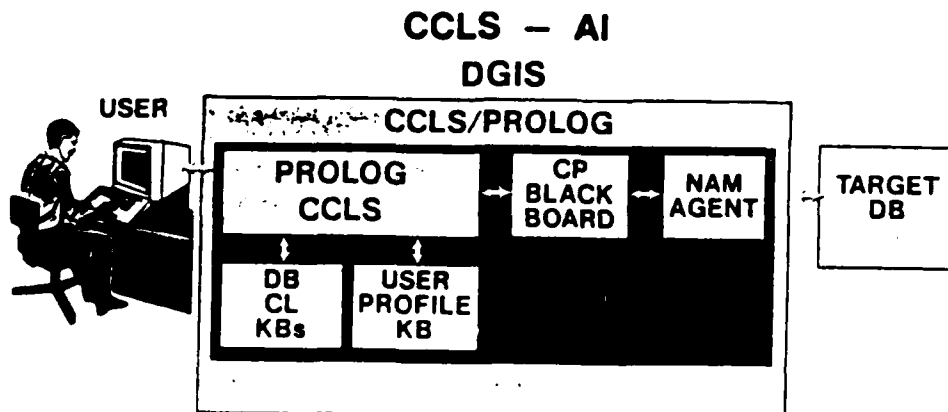
b. Its Database Capability: In that PROLOG has its own internal databases, this feature allows a PROLOG program to manipulate codes as relations that can be asserted or deleted. PROLOG incorporation in CCL includes extending to external databases, e.g., INGRES DBMS in the DGIS software, to achieve the flexibility of storing knowledge in both PROLOG internal databases and traditional external databases. This allows including more powerful database technology in the program system for greater performance and easier use of DGIS by the enduser.

c. The Separation of Logic and Control: A PROLOG program amalgamates rules and facts, basically making one also the other. Although they are governed by a default execution control, the control can be easily supplemented or replaced by more powerful meta-rules also coded in PROLOG.

d. Object Inheritance and Message Passing: These are two powerful features of object language methodology. Both are easily implemented and embedded in PROLOG. Both features are elemental for the more graceful functioning of CCL.

# 2. THE CONCEPTUAL RESTRUCTURING OF CCL: CCLS

Using C language programming, the basic CCL elements consisted of the user, the CCL, the database language processor, and the database information accessed. The jump to PROLOG opened new possibilities in which CCL now could be handled as a knowledge-based system. The CCL evolved into Common Command Language System (CCLS), and its conceptual structure now became:

## CCLS — AI
## DGIS



USER

CCLS/PROLOG

| PROLOG CCLS | CP BLACK BOARD | NAM AGENT |

TARGET DB

| DB CL KBs | USER PROFILE KB |

To the DGIS user making use of CCLS, the fact that it is PROLOG-driven is transparent. The PROLOG CCLS, however, in serving the user, draws on the command language knowledge base, and also a CCLS-user profile knowledge base (in development). The user's query and profile data is controlled through the control program blackboard, which coordinates translation and communications in a continuous real-time system mode [BA-86] through the NAM connection agent. The NAM agent passes the communications to and from the target database system's command language processor for searching on the database information.

With the transition to an AI-based CCL System, the goals of DGIS CCL have been re-constituted to incorporate AI-supported capabilities as follows:

a. One command language to communicate with all bibliographic databases.

b. Creation a CCL System that assists the user in searching unfamiliar database systems.

c. Provision of a user-friendly search session.

d. Provision of an intelligent, user-useful search session.

e. Flexibility to adapt easily to changes and enhancements.

## 3. OTHER CHANGES IN THE ACTIVITY

Because of the relative ease of learning PROLOG programming, another effect of making the transition to PROLOG was to transfer much of that programming from the technical expert to the requirements expert. This change allowed fuller control of the command requirements, from command language researching to command language knowledge base building. This also allowed the technical person to concentrate on the knowledge base - database system connector programs, in itself a programming-intensive activity.

## 4. CURRENT DIRECTIONS

Our next phase in CCLS is a melding of PROLOG implementation, expert system building, and C supplementary programming. The PROLOG-based CCLS has two parts [TDTpip]. One part is fixed, in compiled PROLOG code; the second is variable, in interpretive PROLOG code. The variable part loads and processes information from the two knowledge bases (KB), the command language KB and the user profile KB. Appropriate tools will be incorporated to maintain the KBs (adding, deleting, modifying information). We have procured an artificial intelligence processor system and an expert system building software tool. The processor is networked to the DGIS computer system, and will serve to both develop and maintain AI applications in CCLS and other AI applications on DGIS [KAD87].

We are investigating several schemes for KB organization. In general, we plan to couple PROLOG with a Relational DBMS (RDBMS) where large KBs (most of which are facts) will reside. The technical issue here is the interface between PROLOG and the RDBMS (likely INGRES). We intend to make this interface through SQL (structured query language) so that it can work with any RDBMS, rather than only with INGRES [TDTpip].

Other CCL System application factors are:

a.  CCLS Integration with Other DGIS Functions:  Other DGIS operations
    are potentials for AI applications, with which to link with CCLS.
    One is the DGIS Directory of Online Resources, wherein a user's
    query resources are identified and communicated with automatically
    and simultaneously.  Another is the DGIS postprocessing routines,
    with which the multiple resource responses are automatically
    downloaded, translated, merged, and processed (or tailored), based
    on a one-pass instruction entry with which the user invokes the
    whole process.
b.  Planning Capability:  Includes the preliminary structuring of
    multiple queries and the combining of target databases' result
    sets.
c.  Learning Capability for CCLS:  Employing learning solution paths
    [BA-86] for optimizing the information added to the command
    language KB and the user profile KB.
d.  Migrate to Natural Language:  The NISO and appended CCL will be the
    backbone of the DGIS CCLS, but in migrating to Natural Language
    dialogue, will become transparent in a command language translation
    supporting role.
e.  Provide Simultaneous Database Access Capability:  That is, true
    concurrent connecting with, searching in, and downloading of
    results from multiple systems.

Considering these CCLS application goals, the following technical
aspects that explain our programming approach to the DGIS CCLS interface
are (and verbatim from [TDT87]):

The DGIS CCLS is structured as a knowledge-based system.  CCLS can be
thought of as a black box between the user and the host database.  The
control program (CP) of CCLS is a blackboard-based architecture PROLOG
program that controls the interaction between CCLS agents and the
communication agents (we use the term 'knowledge source' rather than agents
to be consistent with the literature on blackboard systems).  There will be
two types of CCLS knowledge bases.  One is pertinent to the user
information, and the other is the knowledge base about databases.  The user
knowledge base (UKB) system will store information relevant to a particular
user, or a group of users.  Examples of information stored in the UKB are
areas of interest (database names), short-hand (CCLS scripts, aliases, et
al.), user privileges, etc.  This information is needed by the CCLS to
intelligently converse with and interpret commands from the user.  The
database knowledge base (DKB) contains information needed to translate CCLS
commands into host database commands and to understand the returning
results and errors from the database.

The control program (CP) is a typical blackboard-based program.  It is
a PROLOG implementation of an object-oriented system where the blackboard
is nothing more than a general object that registers and monitors progress
of the related knowledge sources.  Each knowledge source is an object that
is activated and deactivated by messages.  A knowledge source's progress
and results are also composed in terms of messages whenever possible for
the blackboard of the CP.  This includes the packaged messages received
through the NAM connection agent, already mentioned.  The following figure
shows the invoking of a CCLS session:

699

```
DGIS/vax[1]% ccl


    *** Welcome to DGIS CCL ***


 This system is the first AI-based component
 on DGIS to give the user standard access to
 diverse information systems.


The following systems are currently available:
  dialog.brs,drols,nasa

Please specify a CCL system (^D to exit) > dialog

Trying to establish connection to DIALOG using NAM.


Attempting hardwired connection at 1200 baud to TYMNET. Connection established to TYMNET.

Attempting TYMNET connection to DIALOG2. TYMNET connection established to DIALOG2.

Attempting login. Login complete.

........
Loading DIALOG knowledge base ...

Loading User Profile Knowledge Base ... [/au/ccl/proto/usertb/kuhn.pl consulted (0.300 sec 340 bytes)]

CCL > explain

The EXPLAIN command provides information on various topics in CCL DIALOG. The following topics
can be explained:

choose  display  find  show  stop
define  explain  general  start

CCL >  etc
```

DGIS CCLS prototype: Initiating DGIS CCL. Note banner, systems available, connecting to target system
in CCLS mode, login, DIALOG Knowledge Base loaded, User Profile Knowledge Base loaded, entry of
NISO common command 'explain' in place of 'help'.


The construction of the CCLS knowledge bases is being done with the aid
of domain experts at DTIC. These experts help in stating the requirements
for interpreting the native command languages, and capture expert searcher
usage of command languages. Additionally, a number of PROLOG tools that
help maintain and validate the knowledge bases will be implemented.

All this to resolve the problem of, as the requirements expert states
[BRL87]: "Ever since the creation of the second online database search
system, the problem of multiple command languages has plagued online
searchers."

## VI. NEXT: DGIS CCLS & HYPERMEDIA

Imagine sitting down at your system, and initiating simultaneous
displays of information from an online database, a CD-ROM database, a file
that you have built yourself that includes figures and graphs, and a high
density video disk that is a general comprehensive reference such as an
encyclopedia. Further imagine having this set of displays include color,
text information, static images, motion imagery, and verbal or music
information that supports the visual information. On top of this, all

interactions and interfaces have been invoked and are controlled through
icons, some of which are programmed into DGIS, some which come from the
media peripherals and some of which you have created yourself.

Hypermedia is the converging of previously separated technologies.
Additionally, bitmap systems are acquiring popularity in the Department of
Defense information community. They serve as standalones either for their
capabilities such as graphics, or as frontends to systems because of their
interface capabilities to existing databases.

In the view of the human user, hypermedia appears in being able to
simultaneously access and display multiple media as sources of information.
This access appears in forms closer to the mind's visualization of
information, rather than straight, passive text. In the hypermedia
environment information can be organized in different ways, depending on
the user's viewpoint. This environment allows the user to organize
information in a visual manner by using the bitmap system capabilities of
icons, multiple windowing techniques, and static and motion imagery.
Hypermedia application allows the user to organize text, pictures, and
sound by association, or context, following the way humans organize
information in their minds. Contextual and spatial cues supplement the
user's model of which information nodes he is viewing, and how those nodes
relate to each other.

Our initial entry into hypermedia concerns looking into AI-based
aspects of hypermedia applications in a desktop environment for CCLS. The
following figure illustrates a composite representation of the desktop
environment [KAD88]:
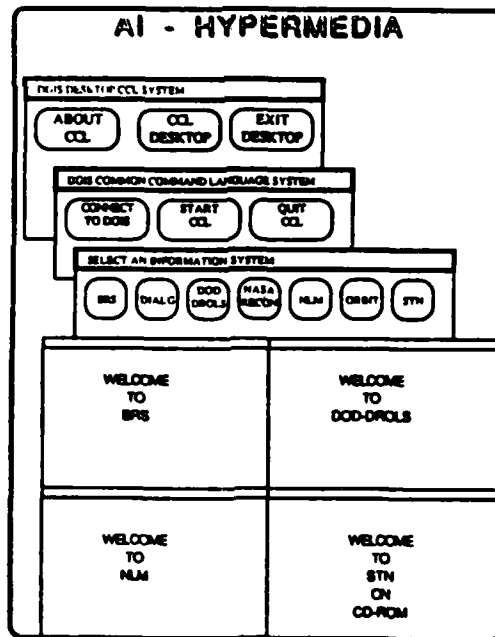


DGIS Entry into Hypermedia: This figure is a composite representation of our initial entry into AI-
Hypermedia. CCLS is being used as the vehicle, merging bitmap system capabilities and PROLOG-based
CCL. We are exploring a desktop CCLS environment, making use of icons, windowing, simultaneous
session displays, and color. A hypermedia link to CD-ROM information bases is a near-term effort. We
are doing this currently on a SUN 3/260 workstation.

CCLS is being used as the vehicle, merging bitmap system capabilities and PROLOG-based CCL. We are currently exploring the desktop on a SUN 3/260 workstation, making use of icons, windowing, simultaneous session displays, and color. A hypermedia link to CD-ROM information bases is foreseen as a near-term development. We feel that getting into the hypermedia environment and looking at the multitude of possibilities will keep us ahead of what is already coming down the road: bitmap systems and hypermedia peripherals in the enduser environment. What we see taking place immediately are:

o   Icon drive -- creating iconic systems to invoke DGIS
    operations.
o   Multiple window displays.
o   Simultaneous tasking -- parallel and disparate tasks
    running concurrently.
o   The use of color -- in that the medium is the message.
o   Multimedia accesses -- online, CD-ROM, video disc,
    other storage media.
o   Intermedia accessing -- simultaneous access and display
    of disparate media.
o   Sound -- verbal/musical information supporting the
    visual information.

We envision CCLS/Hypermedia expanding to a desktop environment for the whole of DGIS, which in itself is evolving into a DoD-wide network of technical libraries, information analysis centers (IACs), and other automated sources of DoD information. This network is provisionally named The Scientific and Technical Information Network (STINET) [CGA87]. As the cost of bitmap systems lowers and their use proliferates, a "STINET Workstation" is forecast, based on the DGIS hypermedia desktop environment. The STINET workstation is to be the window on the worlds of text, video, and sound. It will bridge the gaps between existing diverse media, and provide ways to cope with the massive amounts of information. This workstation is to provide an easy-to-learn and -use user interface for accessing not just heterogeneous information systems, but heterogeneous media-based information resources.

An important element is the development of a hypermedia integrator. The integrator will have the capacity to link, operate, and interface with different media, to provide more comprehensively contextual information. This gives the user the capability to link pieces of his text and non-text files to, for example, dictionaries, encyclopediae on CD-ROM, voices and sounds in digitized formats, data points on graphs, et al. These linkages are envisioned as often being non-linear and non-structured in an intuitive manner, so that the flow is not disrupted because of the presence, difference or absence of components and media [TDT88].

The current DGIS system provides users with the capability to access information in remote databases, download the information into local files, and process that information with supplied tools. The medium available for storing information is text files. Our goal for a hypermedia system is not to replace DGIS as it currently stands, but to co-exist with it. Hypermedia capabilities will serve as a DGIS power tool for those who wish to make use of them [TDT88]. The DGIS Common Command Language System development is our vehicle for evolving toward that hypermedia environment.

702

AUTHOR

Allan D. Kuhn
Defense Technical Informaton Center
(DTIC-EB)
Cameron Station
Alexandria, VA   22304-6145
U.S.A.
Telephone (703) 274-5367

REFERENCES

[BRL87]   Bixby, Randy L.  The DoD Gateway Information System (DGIS):
          Common Command Language Mapping.  Defense Technical Information
          Center, Alexandria, VA.  October 1987, AD-A185 951.
[BA-86]   Bundy, Alan.  Catalogue of Artificial Intelligence Tools.
          Springer-Verlag Berlin Heidelberg.  1986.
[CGA85]   Cotter, Gladys A.  The DoD Gateway Information System.  Defense
          Technical Information Center, Alexandria, VA.  October 1985, AD-
          A161 701.
[CGA87]   Cotter, Gladys A.  The Scientific and Technical Information
          Network (STINET): Foundation for Evolution.  Defense Technical
          Information Center, Alexandria, VA.  September 1987, AD-A189 750.
          Also available in NATO/Advisory Group for Aerospace Research &
          Development (AGARD) Conference Proceedings No. 430, March 1988, as:
          "Information Retrieval Systems Evolve - Advances for Easier and
          More Successful Use."
[KAD86]   Kuhn, Allan D., and Gladys A. Cotter.  The DoD Gateway
          Information System (DGIS): User Interface Design.  Defense
          Technical Information Center, Alexandria, VA.  August 1986,
          AD-A174 150.
[KAD87]   Kuhn, Allan D.  Toward an Artificial Intelligence Environment for
          DTIC: Proposed Tasks, Recommended Configurations, Projected Start-
          Up Costs.  Defense Technical Information Center, Alexandria, VA.
          May 1987, AD-A181 103.
[KAD88]   Kuhn, Allan D., and Gladys A. Cotter.  Microcomputers for
          Information Management: An International Journal for Library and
          Information Services: "The DoD Gateway Information System (DGIS):
          The Department of Defense Microcomputer User's Gateway to the
          World."  Ablex Publishing Corp., Norwood, NJ.  Volume 5, Number 2,
          June 1988, pp. 73-92.
[NISO86/87] National Information Standards Organization (NISO),
          Washington, DC.  American National Standard -- For Information
          Sciences: Common Command Language for Online Interactive
          Information Retrieval (DRAFT).  [March 1987] ANSI Z39.58-198X.

[TDT87]    Tran, Duc T.  DoD Gateway Information System (DGIS) Common
           Command Language: PROLOG Knowledge Base Profile.  Consultant;
           Defense Technical Information Center, Alexandria, VA.
           October 1987, AD-A186 150.
[TDT88]    Tran, Duc T.  [Proposal for] A Hypermedia System for DGIS.
           Consultant; Defense Technical Information Center, Alexandria, VA.
           Internal paper.  April 1988.
[TDTpip]   Tran, Duc T.  DoD Gateway Information System (DGIS):  Prototype
           Programming the DGIS Common Command Language.  Consultant; Defense
           Technical Information Center, Alexandria, VA.  Paper-in-progress.
[UNIXol]   University of California-Berkeley.  UNIX Programmer's Manual.
           LEX(1); YACC(1); NAM(1T).  Online manual, UNIX Operating System
           BSD 4.2.